National Security Agency/Central Security Service

INFORMATION
ASSURANCE
DIRECTORATE

Scripting for Bash Vulnerability/Shellshock

MIT-002R-2014
October 2014

Summary

Security researchers, vendors, and other reporting organizations have commented on the GNU Bash (Bourne Again shell) vulnerability, the severity of the vulnerability, and the critical need to patch vulnerable versions of Bash. Central to their message is the need to test for the vulnerability by issuing the exploit, and then patching the affected systems. This technical report presents an introduction for technical and non-technical managers who are unfamiliar with the Bash vulnerability. In particular, this note introduces a few sample code fragments that can test for the vulnerability without exploiting the vulnerability.

The Bash vulnerability, publicly known as Shellshock, exists in the common Unix®[1]-based systems command line program Bash. Often Bash is the default interface on these Unix systems. The vulnerability leverages unchecked trailing strings after the function definitions in the values of the environmental variables; these strings can be arbitrary code.

Am I vulnerable to Shellshock?

If your system is running Bash version 1.14 through 4.3, it is vulnerable to CVE 2014-7169, which includes CVE 2014-6271, CVE 2014-7169, CVE 2014-7186, CVE 2014-7187, CVE 2014-6277, and CVE 2014-6278.[2] In layman's terms, this vulnerability affects most Unix, Linux®[3], or any *nix based system running a command line, over the past twenty five years.

Although Bash is not native to Windows®[4], ported versions are available via utilities such as Cygwin, GitHub, and others. These ported versions can and have been found to be vulnerable to Shellshock, however the remote attack vector is significantly reduced and/or non-existent as, in general, other Windows applications and software does not depend on or utilize Bash and are unlikely to invoke a vulnerable Bash shell.

How would I know if the vulnerability can even be exploited?

Several public advisories suggest that system owners issue the exploit against their own system, by setting an environmental variable with an open string containing a legitimate command and a wildcard.[5] If you wish to check the version of Bash without issuing the exploit, try something like the sample listed below:

[1] Unix® is a registered trademark of The Open Group.

[2] The range of vulnerable versions is not inclusive for all variants of Unix. Some vendors only update the sub-minor version and others backport the patched version to the existing version number.

[3] Linux® is a registered trademark of Linus Torvalds.

[4] Windows® is a registered trademark of Microsoft Corp.

[5] Although readily found on the Internet, this report is not intended to elicit actual system exploits.

```
#!/bin/bash
VERSION=$(bash --version | grep -m 1 -o -P ‘version.{0,4}’)
echo “Bash may be vulnerable for versions below 4.3; your version is”$VERSION
exit 113
```

This sample script utilizes the command *bash --version*, which if run without the quantifiers of the script, will display the full version, release, copyright, and license information. The sample script would produce the following output, where X and Y are major and minor versions:

```
Bash may be vulnerable for versions below 4.3; your version is version X.Y
```

Is there another approach to retrieve similar information?

Additional information about a system and its version of Bash may be obtained from the system’s package manager. For example, on a system that uses the Yellowdog Updated Modified (YUM), one could execute the following command:

```
$ yum info bash
```

The resulting information can then be cross-referenced with versioning and vendor supplied patch information to determine if the system is vulnerable.

What are some useful commands?

At the command shell ($):

```
$ bash --version              ; echo Check the version of Bash.
$ printenv                    ; echo Print the environmental variables.
$ history                     ; echo List the history of the commands run.
$ strings /bin/bash | grep bash  ; echo An oblique method for Bash version.
```

How can I mitigate the Shellshock vulnerability?

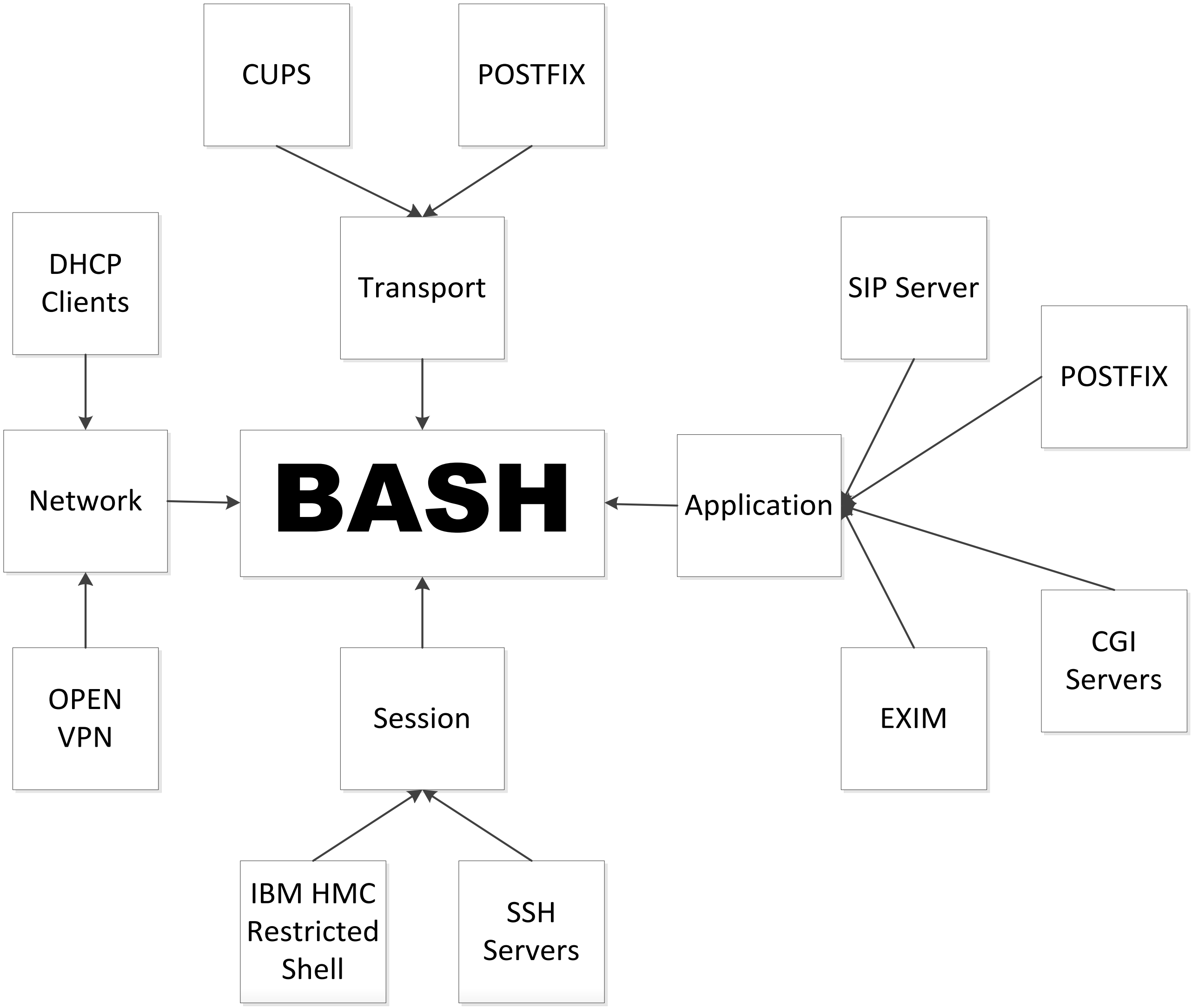
The preferred mitigation is to patch the vulnerable version of Bash and/or upgrade to a non-vulnerable version. It may be necessary to apply subsequent and additional patches for Bash. This may involve installing an update to the operating system and should be tested. Recent developments revealed that early patches needed further correction. It may be necessary to re-visit the status of Bash and this vulnerability in the future. If you cannot patch:

- If feasible, switch out the shell you are using. i.e*. dash, ash, sh, ksh.* Be careful about particular dependencies on Bash that may be affected by a change of shell.

- A more advanced option is to create a custom script to alert on new environmental variables being declared with wild cards.

Possible Shellshock Attack Vectors

The following figure shows examples of possible Shellshock attack vectors. The figure shows a few services that rely on Bash and an associate role (e.g., transport, application, network, session) for the service. Note that this diagram does not include all possible vectors (services and roles). If a service runs on Bash, the possibility exists that it is vulnerable. This demonstrates that the overall attack surface for Bash is more than the command line. Bash is an integral component of the system.



Disclaimer of Endorsement

The guidance in this document is provided "as is." In no event shall the United States Government be liable for any damages, regardless of the nature or theory of liability, arising in any way out of the use of or reliance on this guidance. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

Contact Information

Industry Inquiries
410-854-6091
bao@nsa.gov
USG/IC Customer Inquiries
410-854-4790
DoD/Military/COCOM Customer Inquiries
410-854-4200
General Inquiries
NSA Information Assurance Service Center
niasc@nsa.gov